Amendments to Claims

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims

1. (currently amended) A system of managing data utilizing one or more processors, comprising:

a plurality of map components, each map component having one or more ports for accepting data and for producing data and each map component encapsulating a particular dataflow pattern;

compiler tools for organizing and linking said map components using said ports into ~~a~~ an executable dataflow application; and

an executor for creating and managing data communication among map components in the dataflow application and executing the dataflow application on said one or more processors with data supplied to the system.

2. (original) The system of claim 1, the compiler including tools for visually creating composite components comprising other map components and tools for visually assembling map components into a dataflow application.

3. (original) The system of claim 1, at least one map component having properties determining map component design behavior.

4. (original) The system of claim 1, at least one map component having properties that affect map component execution behavior.

5. (original) The system of claim 1, at least one of the map components comprising a composite component encapsulating a particular dataflow pattern using other map components as subcomponents.

6. (original) The system of claim 1, at least one of the map components comprising a scalar map component to process a specific data transformation.

7. (original) The system of claim 1, at least one of said ports linked to transfer specific types of data.

8. (original) The system of claim 1, at least one of said ports initially defined as a generic port for processing generic types of data, said generic port being later synthesized to transfer a specific sub-type of data.

9. (original) The system of claim 1, at least one of said ports being composite, comprising a plurality of hierarchical ports.

10. (original) The system of claim 1, at least one of said ports supporting multi-valued null data tokens.

11. (currently amended) The system of claim 1, at least one of said map components being encoded as an encrypted ~~XML~~ extensible markup language (XML) document.

12. (original) The system of claim 1, at least one of said map components being composite comprising a number of hierarchical dataflow graphs.

13. (original) The system of claim 1, the compiler operating to remove design time links between map components to produce a flat dataflow graph containing a plurality of map processes for execution.

14. (original) The system of claim 1, the executor operating to assign a thread to each map process for parallel execution.

15. (original) The system of claim 1, the compiler tools operating to perform syntactic and semantic analysis, type inference and validation.

16. (original) A method of transforming data in parallel processing environments where map components are assembled visually into an integrated dataflow application by linking the map components and the integrated dataflow application is executed in parallel by recognizing the linked processes within the map components and allocating a thread to each process.

17. (original) The method of claim 16, wherein a plurality of map processes read data tokens from input ports and write data tokens to output ports.

18. (original) A method of managing data comprising:

accessing a library of map components at least some of said map components constituting a specific data transformation and having input and output ports;

assembling a dataflow application using map components from said library linked together using said ports; and

executing the assembled dataflow application with source data.

19. (original) The method of claim 18, including imposing properties on the map components during assembly constraining the assemblage of the dataflow application.

20. (original) The method of claim 18, the map components including polymorphic ports which declare status as input and output ports during assemblage.

21. (new) The system of claim 14, the executor operating on a single CPU in a hyperthread architecture.

22. (new) The system of claim 14, the executor operating on a multiple processor core with at least some threads assigned to different processors.

23. (new) The system of claim 14, the executor operating on multiple processors in a distributed network configuration.